Compact 32-bit CMOS adder in multiple-output DCVS logic for self-timed circuits

G.A.Ruiz and M.A.Manzano

Abstract: The paper presents a compact 32-bit carry look-ahead (CLA) adder in multiple-output differential-cascode voltage-switch (MODCVS) logic for delay-insensitive self-timed applications. This adder is structurally and functionally equivalent to a dynamic Manchester carry chain with an efficient organisation which exploits the advantages of MODCVS logic to reduce both the number of devices required and the routing area. The electrical simulation carried out on a standard CMOS 1.0μm design shows that this adder is similar in speed to the binary carry look-ahead adder previously reported, though it has a slightly higher average addition time. However, the MODCVS adder occupies 50% of the area, uses 36% fewer transistors and has 20% less dynamic power consumption. On comparing it with similar asynchronous adders, it minimises the worst-case delay, maintaining similar average delay, making it suitable in circuits with nonrandom input operands.

1 Introduction

Self-timed circuits are one possible solution to clock distribution problems in VLSI circuits [1]. The claimed advantages are that such circuits are capable of operating at the maximum speed determined by the intrinsic hardware delays (they can be designed for average-case rather than worst-case performance), they have no problems with clock skew and they may have a substantial low-power advantage over clocked circuits [2]. Different prototypes of self-timed processors [3–5] developed in academic research have proved the feasibility of self-timed designs. Recently, a commercial implementation of a self-timed DSP [6] has demonstrated the advantage of the self-timed design over its synchronous counterpart in power- and noise-sensitive applications.

The 'delay-insensitive' self-timed circuits use a handshaking protocol with a dual-rail data encoding to make them independent of both circuit and interconnection delays. The dynamic differential-cascode voltage-switch (DCVS) logic presents a dual-rail coded nature very popular in the literature on self-timed processors [1, 3, 7]. This logic requires true and complement input signals to switch the two outputs to different logic states; this makes it a complete logic family capable of generating any logic expression in a simple way [8]. This advantage, in addition to its inherent self-checking capacity [9], has led to the development of commercial design and analysis tools for reliable/fault-secure circuits based in DCVS logic [10].

Different types of asynchronous adders carried out in DCVS logic have been described in [11–14]. These adders, together with other asynchronous adders [15, 16], assume a random distribution of input operands where the average carry propagation length in an n-bit asynchronous binary addition can be closed to $\log_2(5n/4)$ [17]. Recently, new architectures of asynchronous adders which are not delay-insensitive with average delay inferior to the above value have been presented in [18, 19]. However, an internal study at the University of Manchester [20] on the addition performed in the asynchronous ARM processor [4] has demonstrated that data operations have carry propagation paths greater than those that might be expected from purely random operands. In these circumstances, asynchronous addition takes a long time.

In this paper, a compact 32-bit CLA adder in multiple-output DCVS (MODCVS) logic with completion circuit for high-speed applications in delay-insensitive self-timed circuits is presented. The adder is based on a dynamic Manchester carry chain, similar to that presented in [21], with an organisation of 2- and 8-bit group generate and propagate terms. The combination of an efficient organisation and compound MODCVS gates allows the circuit's device count to be reduced and a saving to be made in routing area since the number of interconnecting lines decreases. The layout of this adder has been made in a standard CMOS 1.0μm technology to compare the results with those presented in [14]. The electric simulation shows that its speed is similar to that of binary carry look-ahead, but this adder is more efficient in area and dynamic power consumption. Comparisons with other similar delay-insensitive asynchronous adders show that this adder's worst-case delay is far smaller, maintaining similar average delay values.

2 CLA in MODCVS logic

DCVS logic is a differential-logic style derived from domino logic made up of two complementary NMOS logic trees, as shown in the basic circuit in Fig. 1. In a DCVS gate, when R is low (precharge phase), nodes Q and $\overline{Q}$ are precharged to high (outputs F and $\overline{F}$ to low) by means of the PMOS transistors. When R is high (evaluation phase), the input lines in the NMOS tree are evaluated by switching one of the outputs to low. The completion signal (Comp) for handshaking requirements in self-timed circuits

IEE Proceedings online no. 20000381

DOI: 10.1049/ip-cds:20000381

Paper first received 26th April 1999 and in revised form 20th January 2000

The authors are with the Departamento de Electronica y Computadores, Facultad de Ciencias, Universidad de Cantabria, Avda. de los Castros s/n, 39005 Santander, Spain

can be generated by a NAND gate connected to nodes Q and $\overline{Q}$. The similarity between DCVS and domino logic allows the application of some of the concepts established for the latter. It is possible to introduce, by extension, the design style called multiple-output DCVS (MODCVS) logic. As a result, the MODCVS logic can reduce the number of stages and transistors, resulting in a shorter circuit delay and a smaller chip area, although in a similar way to MODL it depends heavily on the degree of recursiveness of the circuits.

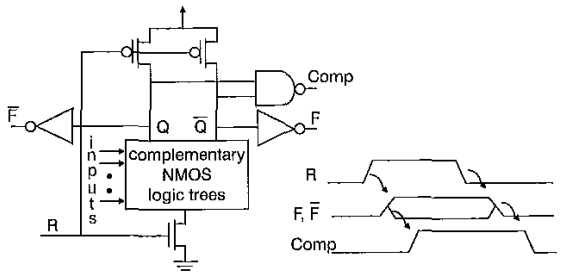

Fig. 1 *DVCS logic with completion circuit and timing*

In [22], an enhanced CLA in MODCVS logic which improves circuit performance and decreases power is presented. In this CLA, if (C_{i-1}, $\overline{C}_{i-1}$) are the carry-in and its complement for stage i, then the carry-out and its complement (C_i, $\overline{C}_i$) can be expressed as

$$\begin{aligned} C_i &= G_i + P_i C_{i-1} \\ \overline{C}_i &= N_i + P_i \overline{C}_{i-1} \end{aligned} \tag{1}$$

where

$$\begin{aligned} G_i &= A_i B_i \\ N_i &= \overline{A_i + B_i} \\ P_i &= A_i \oplus B_i \end{aligned} \tag{2}$$

are the generate signals, G_i and N_i, and the propagate signal P_i. Expanding this yields

$$\begin{aligned} C_i &= G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \ldots \\ &\quad + P_i P_{i-1} \ldots P_1 C_0 \\ \overline{C}_i &= N_i + P_i N_{i-1} + P_i P_{i-1} N_{i-2} + \ldots \\ &\quad + P_i P_{i-1} \ldots P_1 \overline{C}_0 \end{aligned} \tag{3}$$

The sum is generated by

$$\begin{aligned} S_i &= C_{i-1} \oplus P_i \\ \overline{S}_i &= \overline{C}_{i-1} \oplus P_i \end{aligned} \tag{4}$$

3 Compact 32-bit CLA adder in MODCVS logic

The eqns. 1–4 allow the implementation in MODCVS logic of a highly compact 32-bit CLA adder organised as shown in Fig. 2. This adder is made up of three cell types identified as CI, CII and CIII. CI is an 8-bit wide cell which produces the 1-bit propagate terms (P_i, $\overline{P}_i$), carry terms (C_i, $\overline{C}_i$) and 8-bit group propagate (PPP_{n+7}) and generates (GGG_{n+7}, NNN_{n+7}) terms (n is 1, 9, 17 or 25), from the input data (A_i, $\overline{A}_i$, B_i, $\overline{B}_i$), carry-in (C_0, $\overline{C}_0$) and the three most significant 8-bit group carries (C_8, $\overline{C}_8$, C_{16}, $\overline{C}_{16}$, C_{24}, $\overline{C}_{24}$). There are four CI-type cells in the adder. Cell CII is an 8-bit group-enhanced CLA gate (Fig. 3) which computes the most significant 8-bit group carries (C_{n+7}, $\overline{C}_{n+7}$) from the carry-in and the 8-bit group propagate and generate terms. Finally, cell CIII is an EXOR gate which generates the sum outputs (S_i, $\overline{S}_i$) from the 1-bit propagate (P_i, $\overline{P}_i$) and carry (C_{i-1}, $\overline{C}_{i-1}$) terms; there are 32 CIII cells in the adder like that shown in Fig. 4, which implement eqn. 4. The generation of the partial-completion signals of each of the 32 sum outputs ($Comp_i$) takes place in CIII, and that of the carry-out completion signal ($Comp_{33}$) in CII; these signals are used subsequently in the generation of a global completion signal.

Cell CI consists of three levels of logic stages, as shown in Fig. 5; the index n represents the least significant bit of the 8-bit group input data. The first level of CI is made up of the CI.1 gates which produce 1- and 2-bit generate and propagate terms from the 2-bit group input data. If the first gate, whose inputs are (A_n, $\overline{A}_{n+1}$), ($\overline{A}_n$, $\overline{A}_{n+1}$), (B_n, $\overline{B}_{n+1}$) and ($\overline{B}_n$, $\overline{B}_{n+1}$), then the 1-bit generate signals are defined as

$$\begin{aligned} G_n &= A_n B_n \\ N_n &= \overline{A_n + B_n} \\ G_{n+1} &= A_{n+1} B_{n+1} \\ N_{n+1} &= \overline{A_{n+1} + B_{n+1}} \end{aligned} \tag{5}$$

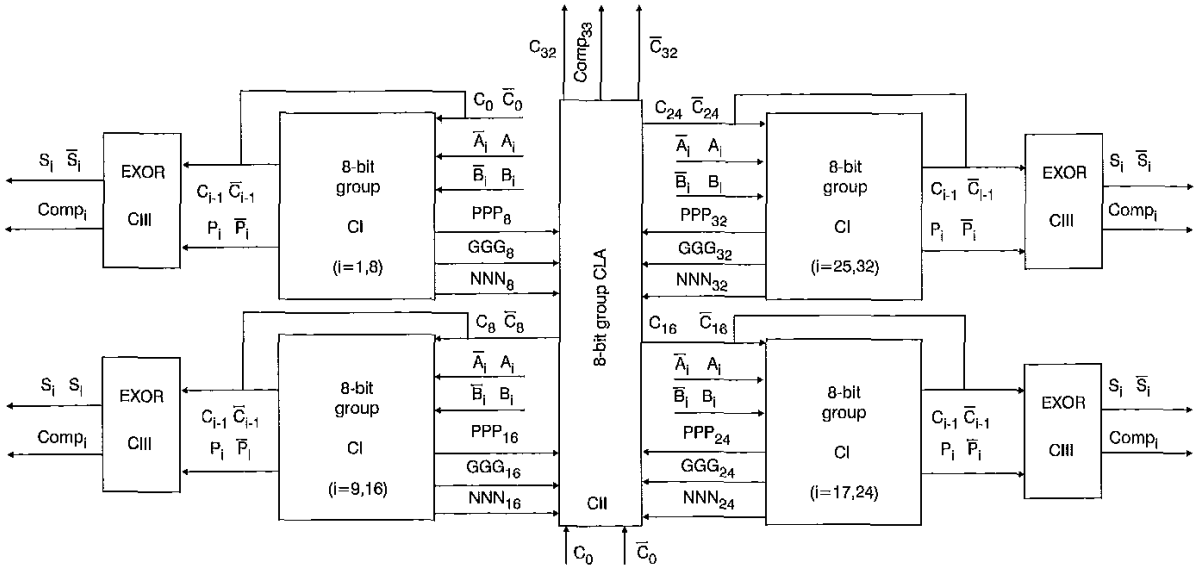

Fig. 2 *Organisation of 32-bit CLA MODCVS adder*

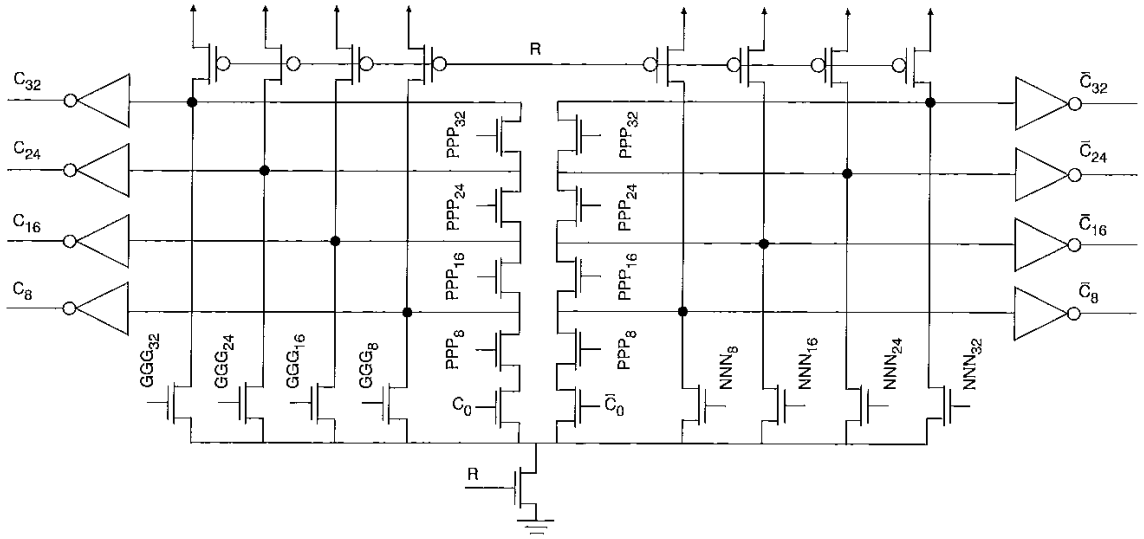

Fig. 3 *Enhanced CLA in MODCVS logic (cell CII)*

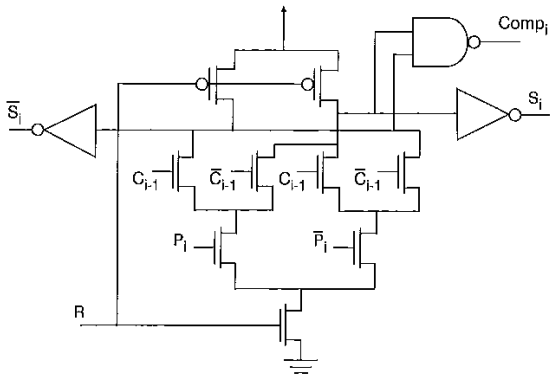

Fig. 4 *EXOR gate (cell CIII)*

the 1-bit propagate signals are defined as

$$P_n = A_n \oplus B_n$$
$$P_{n+1} = A_{n+1} \oplus B_{n+1}$$
$$\overline{P}_n = \overline{A_n \oplus B_n} = G_n + N_n$$
$$\overline{P}_{n+1} = \overline{A_{n+1} \oplus B_{n+1}} = G_{n+1} + N_{n+1} \qquad (6)$$

and the 2-bit generate and propagate signals are defined as

$$GG_{n+1} = G_{n+1} + P_{n+1}G_n$$
$$PP_{n+1} = P_{n+1}P_n$$
$$NN_{n+1} = N_{n+1} + P_{n+1}N_n \qquad (7)$$

The high degree of recurrence of these signals allows eqns. 5–7 to be implemented in a single complex gate, as shown in Fig. 6; note that, with the signals N_n, N_{n+1}, G_n, G_{n+1}, P_n, P_{n+1}, the rest of the signals can be generated. This complex gate is a compound MODCVS gate where some inverters have been replaced by other inverting static CMOS gates to achieve more efficient implementation of logic functions. As a result, the proposed cell CI.1 reduces the number of transistors from the 86 of a conventional scheme to 60. This reduction in the device count is important, as there are 16 CI.1 cells in the whole adder.

The second level of the CI is made up of gates CI.2 and CI.3 which produce the 4-bit group generate (GGG_{n+3}, NNN_{n+3}) and propagate (PPP_{n+3}) terms and 8-bit group generate (GGG_{n+7}, NNN_{n+7}) and propagate (PPP_{n+7}) terms, from 2-bit generate and propagate terms. These are

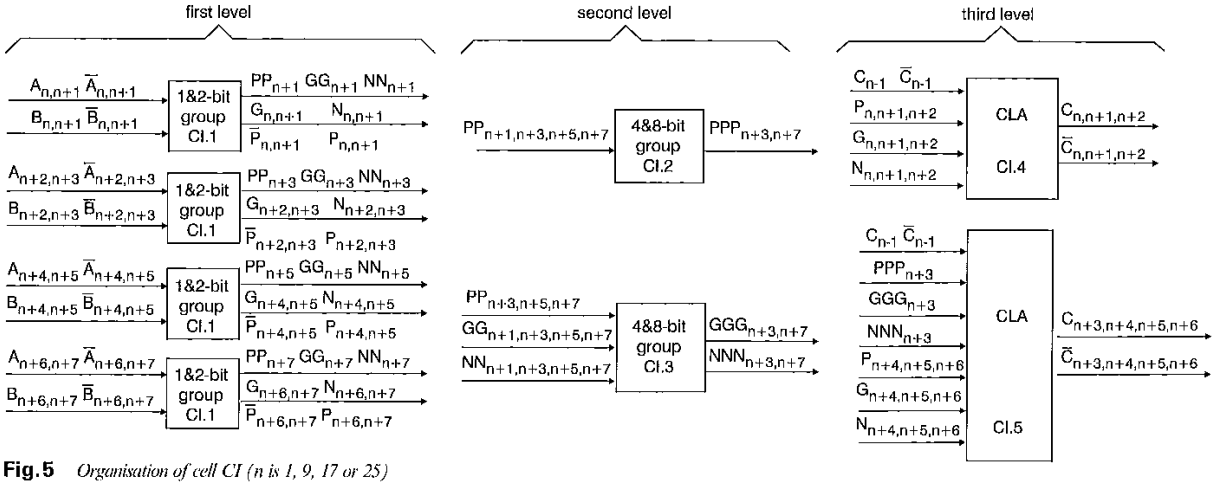

Fig. 5 *Organisation of cell CI (n is 1, 9, 17 or 25)*

defined as

$$
\begin{aligned}
PPP_{n+3} &= PP_{n+3}PP_{n+1} \\
PPP_{n+7} &= PP_{n+7}PP_{n+5}PPP_{n+3} \\
GGG_{n+3} &= GG_{n+3} + PP_{n+3}GG_{n+1} \\
GGG_{n+7} &= GG_{n+7} \\
&\quad + PP_{n+7}(GG_{n+5} + PP_{n+5}GGG_{n+3}) \\
NNN_{n+3} &= NN_{n+3} + PP_{n+3}NN_{n+1} \\
NNN_{n+7} &= NN_{n+7} \\
&\quad + PP_{n+7}(NN_{n+5} + PP_{n+5}NNN_{n+3})
\end{aligned} \tag{8}
$$

Figs. 7 and 8 show the implementation in multiple-output logic of eqn. 8. CI.2 is a MODL gate (Fig. 7) which produces the propagate signals and CI.3 is a MODCVS gate (Fig. 8) which produces the generate signals. This latter gate can be implemented in MODCVS since $PP_sGG_s = 0$ and $PP_sNN_s = 0$ ($s = n + 3, n + 5, n + 7$), thus avoiding the problems of 'false' discharges in the dynamic output nodes.

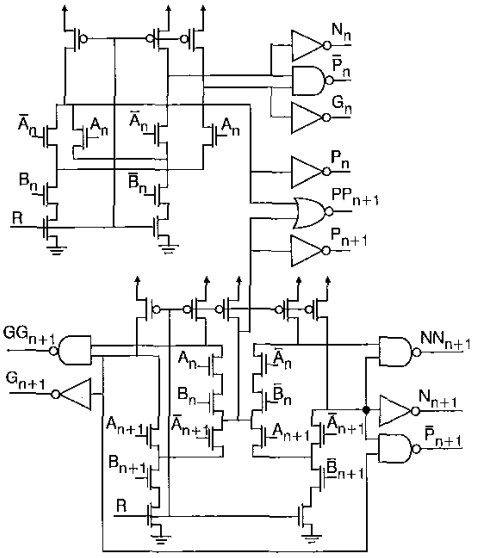

Fig. 6 *Implementation of CI.1*

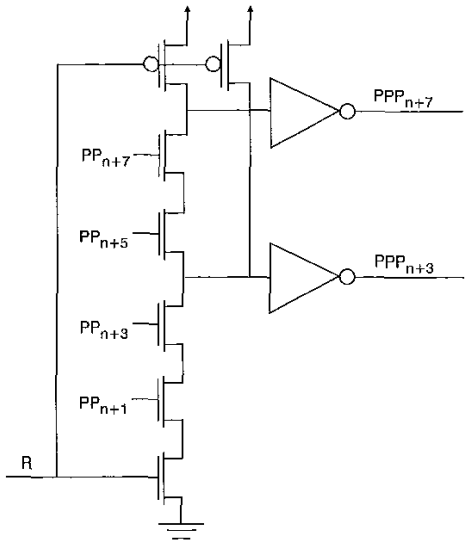

Fig. 7 *Implementation of CI.2*

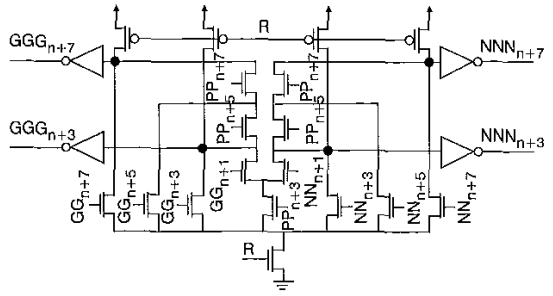

Fig. 8 *Implementation of CI.3*

The third and final level of cell CI is formed by CI.4 and CI.5 which are 3-bit and 4-bit enhanced CLA gates, respectively. In this level, the seven output carries are computed from the 1-bit and 4-bit group-generate and propagate terms, and the 8-bit group carry generated in CII.

4 Results and comparative analysis

The 32-bit MODCVS adder with completion circuit proposed in this paper was designed using a standard double-metal 1.0μm CMOS technology. It was simulated with the HSPICE electrical simulator at level 6, 25°C, VDD = 5.0V and CL = 0.2pF using fast, typical and slow device technology parameters. This adder includes distributed clock buffers which introduce a delay of 0.7ns (fast), 1.1ns (typical) and 1.4ns (slow). This technology is identical to that used in [14] to implement three self-timed adders: a 32-bit ripple-carry (RC) adder, a 32-bit carry look-ahead (CLA) adder, and a 32-bit binary carry look-ahead (BCL) adder. Table 1 lists the main parameters of this MODCVS adder, which are evaluated and compared with those outlined in [14]. The mask layout of the MODCVS adder has a final size of 530 × 2200μm^2 and a total number of transistors of approximately 2100. This adder has an area 44% greater and has 20% more transistors than the CLA adder, but its size is far less than the relatively large BCL adder which occupies an area 100% greater and has 56% more transistors.

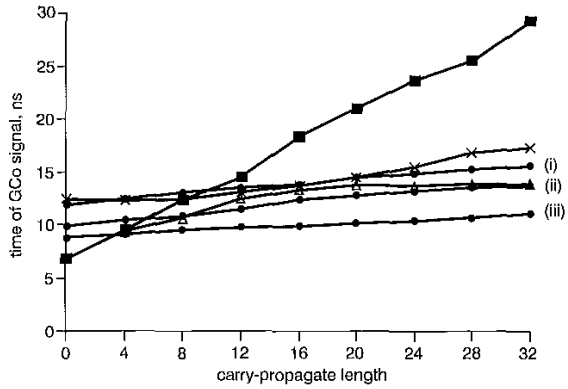

Fig. 9 *Addition time for different carry propagate lengths at VDD = 5V*
(i) Slow; (ii) typical; (iii) fast
—●— MODCVS adder
-·■-· RC adder
—×— CLA adder
--^-- BCL adder

Fig. 9 shows the addition time for different carry-propagate lengths; the average carry propagation length is $\log_2(5n/4) = 5.32$. In [14], it was demonstrated that the BCL adder is the fastest for any input-data condition. The MODCVS adder is, using typical parameters, even faster

Table 1: Comparison between this MODCVS adder and adders presented in [14]

	Adders from [14]			This work		
	RC adder	CLA adder	BCL adder	MODCVS adder		
Area (μm2)	274 × 2430	304 × 2667	1020 × 2265	530 × 2200		
No. of transistors	1525	1745	3271	2100		
				Fast	Typ	Slow
Average addition time (ns)	10.5	12.3	9.8	9.3	10.6	12.6
Worst-case addition time (ns)	29.2	17.3	13.8	11.1	13.7	15.6
Dynamic power consumption (min/max) at 10 MHz (mW)	36.8/41.8	46.6/49.3	74.1/79.3	66.4/ 72.9	58.8/ 63.5	53.0/ 56.4

than the BCL adder for certain carry-propagate lengths. However, the average addition time (10.6ns for 'typical') is greater than that of the BCL adder (9.8ns) and similar to that of the RC adder (10.5ns). The worst-case delay takes place between the carry-in and higher output sum with a delay of five logic stages with the following worst-case path: CI.1, CI.3, CII, CI.5 and CIII; their values are 11.1ns (fast), 13.7ns (typical) and 15.6ns (slow). The carry-out has a worst-case path of three logic stages: CI.1, CI.3 and CII. A 64-bit adder made up of two 32-bit adders in ripple carry configuration has an average addition time of 10.4ns (fast), 11.7ns (typical) and 13.7ns (slow), and a worst-case addition time of 13.2ns (fast), 16.9ns (typical) and 19.6ns (slow). Finally, Fig. 10 shows the distribution curves of a

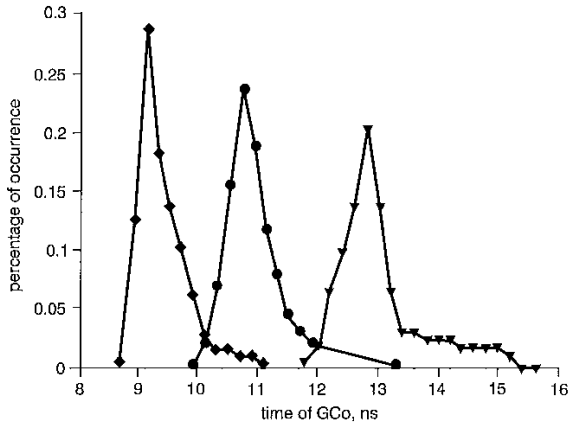

Fig. 10 *Distribution of GCo signal*
◆ fast
● typical
▼ slow

global completion signal (GCo) for the different device parameters whose maximum values represent the average addition time. These curves have very narrow limits which indicate the slight incidence of the carry propagation length on the delay of addition, a property which can be taken advantage of in asynchronous circuits where the focus is on data-independent parameters.

The completion circuit described in [14] is identical in all the adders and generates a GCo which indicates when the addition process has finished. The addition time is measured from when signal R goes to high until the GCo goes to high and depends on carry propagation length. The completion circuit adds extra overhead and delay in the adder. In the worst case, when all completion signals are generated simultaneously, the completion circuit shows a typical delay time of 3.7ns. However, with random input operands these signals are generated at different instants in time so that the average delay of the completion circuit can be estimated at 2ns.

The MODCVS adder uses a 'pure' delay-insensitive approach preferred in circuit compilation and automatic circuit layout tools, unlike other similar asynchronous adders where the completion circuit is driven by carry signals rather than sum signals. In these latter adders, the completion generation can start earlier in parallel with the last-sum generation (whose delay is a constant), increasing the speed of addition. This approach relies on the 'safe' assumption that sum generation is faster than completion generation. Typical examples of this type of not strictly delay-insensitive adders are described in [13, 20]. Other speed-efficient asynchronous adders [15, 19, 23] based on a non-delay-insensitive approach assume that the delay in all circuit elements and wires is known (or at least bounded).

Table 2: Comparison with other CMOS asynchronous adders

	Size	Technology (μm)	t_{ave} (ns)	t_{wc} (ns)	t_{wc}/t_{ave}
'Pure' delay-insensitive					
[24]	32-bit	1.6	11	40	3.6
[This work (typical)]	32-bit	1.0	10.6	13.5	1.3
Not strictly delay-insensitive					
[20]	32-bit	1.2	12	21	1.8
[13] RCA	32-bit	1.2	8.93	27.96	3.1
CLA-2 bit	32-bit	1.2	6.33	18.66	2.9
Not delay-insensitive					
[19]	56-bit	1.0	1.81	—	—
[23]	56-bit	0.5	1.28	—	—

Size= length of adder. Technology = CMOS technology
t_{ave} = average addition time
t_{wc} = worst-case addition time

These circuits are more difficult to design because these adders apply worst-case models in both input data and physical properties when inserting delays, thus leading to worst-case behaviour. Moreover, their testing is very complicated. Table 2 shows the average (t_{ave}) and worst-case (t_{wc}) delays of some asynchronous adders which use different delay assumptions. The proposed MODCVS adder presents the best t_{wc} in comparison with other adders; the ratio t_{wc}/t_{ave} is the best in all cases. This means that this adder is recommendable in delay-insensitive asynchronous circuits with nonrandom input operands. Moreover, the t_{ave} is similar to that in [24], although the technologies are different. To compare this adder with not strictly delay-insensitive adders, t_{ave} the buffer clock time (1.1ns) and the last EXOR sum gate (1.4ns) must be subtracted from t_{ave}, resulting in 8.1ns, which is only improved on in the CLA-2 bit of [13]. In all cases, it can be observed that the non-delay-insensitive adders present the best high-speed performance since they have a far lower t_{ave} than the rest of the adders; more specifically, the 56-bit carry-select adder based on the statistical carry look-ahead adder proposed in [19] has a t_{ave} of 1.28ns at $V_{DD} = 3.3$V.

5 Conclusions

A compact 32-bit CLA adder in MODCVS logic with completion circuit for delay-insensitive self-timed applications is presented. This adder uses an efficient organisation and compound MODCVS gates to reduce both the number of devices and routing area. Its structure can be extended to other differential logic (the interested reader should refer to [25]). On comparing the results with the adders presented in [14], the MODCVS adder is found to be highly efficient in terms of area and power with respect to the high-speed BCL adder, preserving similar speed characteristics. Moreover, it has the best t_{wc} in comparison with other delay-insensitive adders, maintaining similar values of t_{ave} although these times are far higher than those obtained for non-delay-insensitive asynchronous adders. The distribution of the GCo signal indicates that the adder can be used for high-speed arithmetic in self-timed circuits where the delay of addition is kept within a narrow range of values independent of input operands (data-independent). Finally, the MODCVS logic, similar to DCVS logic, is fault-secure by design, i.e. in the presence of a single fault, a block of MODCVS logic either produces the correct output for the given inputs or produces an output that is detectably different. A simple check circuit can be placed on the circuit outputs at very low overhead to provide online, real-time error detection [9, 10].

6 References

1 MENG, T.H.: 'Synchronization design for digital system' (Kluwer Academic Publishers, Boston/Dordrecht/London, 1991)

2 HAUCK, S.: 'Asynchronous design methodologies: an overview', *Proc. IEEE*, 1995, **10**, (1), pp. 69-93

3 JACOBS, G.M., and BRODERSEN, R.W.: 'A fully asynchronous digital signal processor using self-timed circuits', *IEEE J. Solid-State Circuits*, 1990, **25**, (6), pp. 1526–1537

4 WOODS, J.V., DAY, P., FURBER, S.B., GARSIDE, J.D., PAVER, N.C., and TEMPLE, S.: 'AMULET1: an asynchronous ARM microprocessor', *IEEE Trans. Comput.*, 1997, **46**, (4), pp. 385–398

5 FURBER, S.B., GARSIDE, J.D., RIOCREUX, P., TEMPLE, S., DAY, P., LIU, J., and PAVER, N.C.: 'AMULET2e: an asynchronous embedded controller', *Proc. IEEE*, 1999, **87**, (2), pp. 243–256

6 PAVER, N.C., DAY, P., FANSWORTH, C., JACKSON, D.I., LIEN, W.A., and LIU, J.: 'A low-power, low noise, configurable self-timed DSP', *Cogency Technol.*, Oct. 1997, (available via http://www.cogency.com)

7 RENAUDIN, M., HASSAN, B.E., and GUYOT, A.: 'A new asynchronous pipeline scheme: application to the design of a self-timed ring divider', *IEEE J. Solid-State Circuits*, 1996, **31**, (7), pp. 1001–1013

8 CHU, K.M., and PULFREY, D.L.: 'Design procedures for differential cascode voltage switch circuits', *IEEE J. Solid-State Circuits*, 1986, **SC–21**, (6), pp. 1082–1087

9 KANOPOULUS, N., PANTZARTZIS, D., and BARTRAM, F.R.: 'Design of self-checking circuits using DCVS logic: a case study', *IEEE Trans. Comput.*, 1992, **41**, (7), pp. 891–896

10 'Design and analysis tools for reliable/fault secure circuits'. Research Triangle Institute. Open Door Program of Mentor Graphics. (information available via http://www.microtec.com/partners/opendoor/research.html)

11 FRANKLIN, M.A., and PAN, T.: 'Performance comparison of asynchronous adders'. Proceedings of IEEE international symposium on *Advanced research on asynchronous circuits and systems*, Salt Lake City, Utah, USA, November 1994, pp. 117–125

12 RENAUDIN, M., and EL HASSAN, B.: 'The design of fast asynchronous adders and their implementation using DCVS logic'. Proceedings of ISCAS 94, London, May 1994, pp. IV.291–IV.294

13 JOHNSON, D., and AKELLA, D.V.: 'Design and analysis of asynchronous adders', *IEE Proc., Comput. Digit. Tech.*, 1998, **145**, (1), pp. 1–8

14 RUIZ, G.A.: 'Evaluation of three 32-bit CMOS adders in DCVS logic for self-timed circuits', *IEEE J. Solid-State Circuits*, 1998, **33**, (4), pp. 604–613

15 NOWICK, S.M.: 'Design of a low-latency asynchronous adder using speculative completion', *IEE Proc., Comput. Digit. Tech.*, 1996, **143**, (5), pp. 301–307

16 RAMACHANDRAN, R., and LU, S.L.: 'Efficient arithmetic using self-timing', *IEEE Trans. Very Large Scale Integr. Syst.*, 1996, **4**, (4), pp. 445–454

17 HENDRICKSON, H.C.: 'Fast high-accuracy binary parallel addition', *IRE Trans. Electron. Comput.*, 1960, **9**, pp. 465–469

18 MULLER, J., TISSERAND, A., and VINCENT, J.: 'Asynchronous sublogarithmic adders'. Proceedings of 1997 IEEE Pacific Rim conference on *Communication, computers and signal processing* (PACRIM97), Victoria, Canada, August 1997, pp. 515–518

19 DE GLORIA, A., and OLIVIERI, M.: 'Statistical carry lookahead adders', *IEEE Trans. Comput.*, 1996, **45**, (3), pp. 340–347

20 GARSIDE, J.D.: 'A CMOS VLSI implementation of an asynchronous ALU'. Proceedings of the IFIP conference on *Asynchronous design methodologies*, Manchester, UK, April 1993, pp. 181–192

21 HWANG, I.S., and FISHER, A.L.: 'Ultrafast compact 32-bit CMOS adders in multiple-output domino logic', *IEEE J. Solid-State Circuits*, 1989, **24**, (2), pp. 358–369

22 RUIZ, G.A.: 'Compact four bit carry look-ahead CMOS adder in multi-output DCVS logic', *Electron. Lett.*, 1996, **32**, (17), pp. 1556–1557

23 CORSONELLO, P., PERRI, S., and COCORULLO, G.: 'Hybrid carry-select statistical carry look-ahead adder', *Electron. Lett.*, 1999, **36**, (7), pp. 549–551

24 MARTIN, A.J.: 'Asynchronous datapaths and the design of an asynchronous adder', *Formal Meth. Syst. Des.*, 1992, **1**, (1), pp. 119–137

25 NG, P., BALSARA, P.T., and STEISS, D.: 'Performance of CMOS differential circuits', *IEEE J. Solid-State Circuits*, 1996, **31**, (6), pp. 841–846